

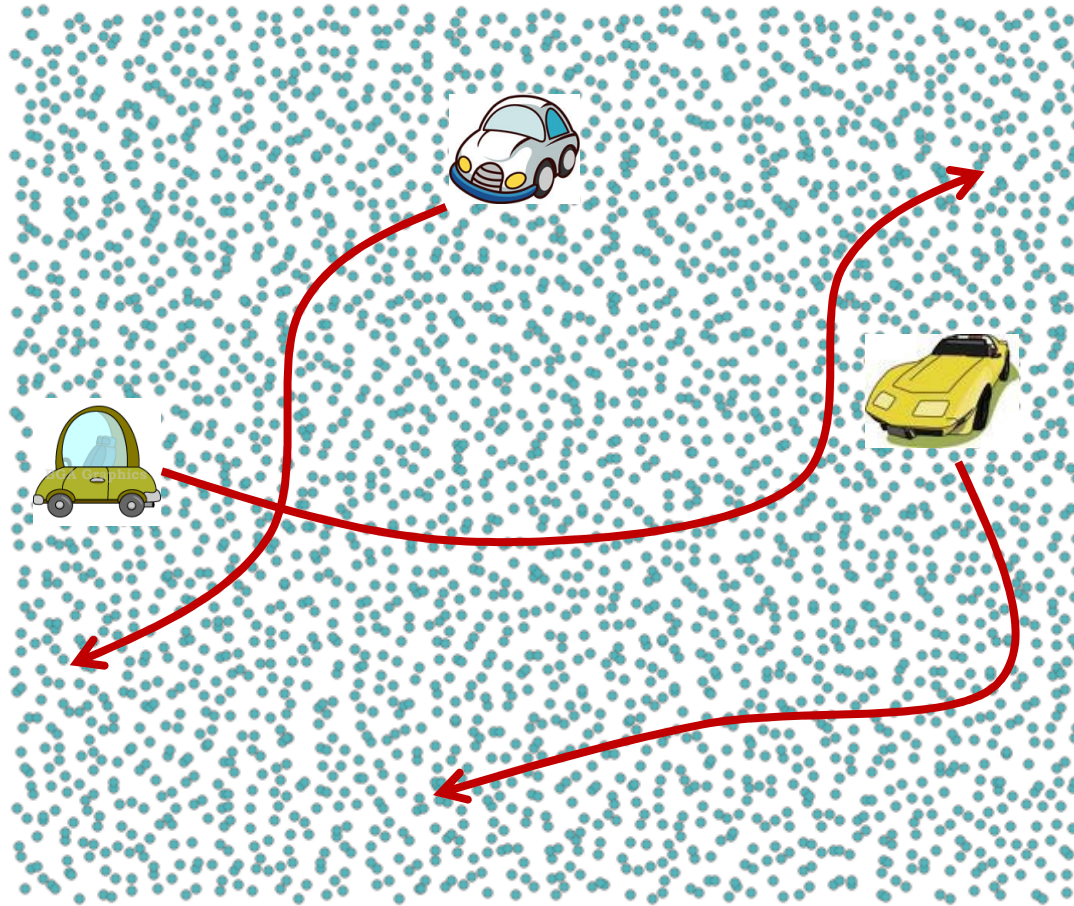
Differential Forms for Target Tracking and Aggregate Queries in Distributed Networks

Rik Sarkar

Jie Gao

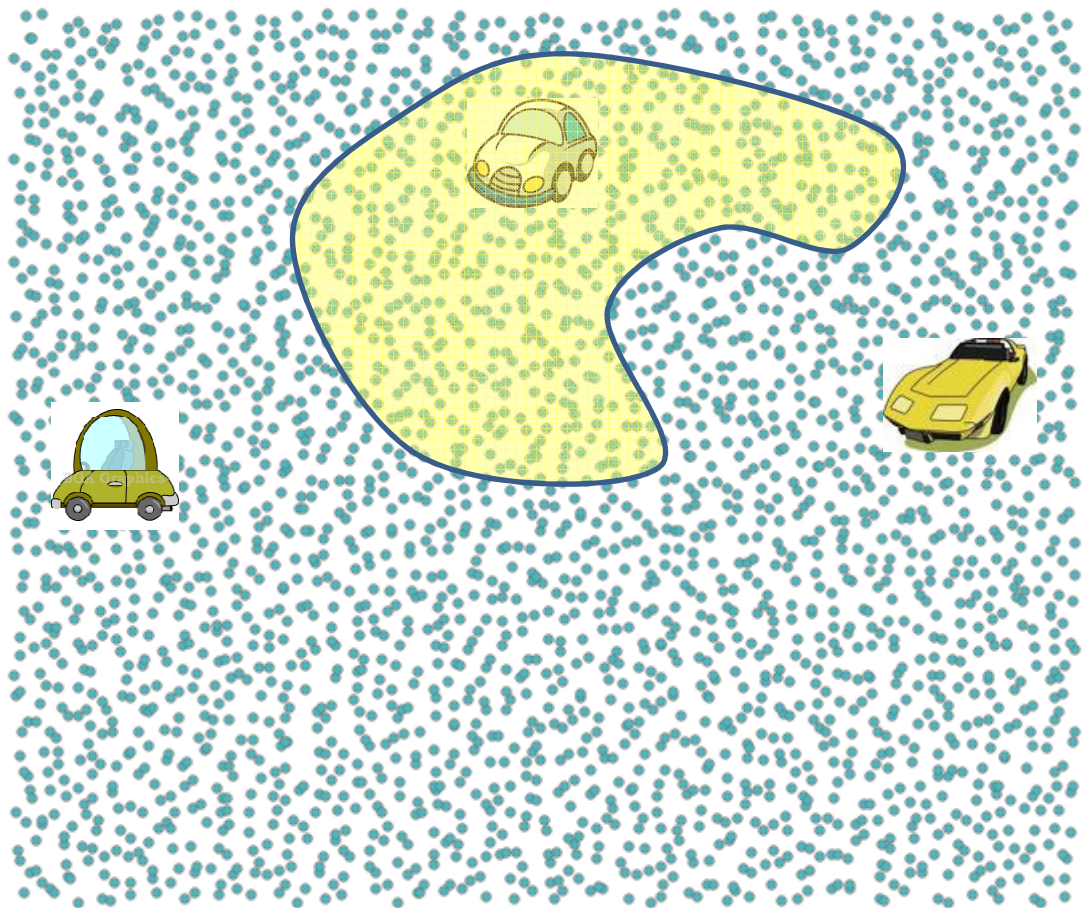
Stony Brook University

Target Tracking with Sensor Networks



Target Queries

- **Range queries:**
targets within any geographic range R .
- **Tracking queries:**
Find the yellow car.
- **Closest target queries**



Naïve Solutions for Range Queries

- Sensors report targets to a central station.
 - Bottleneck and single point of failure.
 - Update cost is high.
 - Query cost is high if the central station is far away.
- Flood the region R , count # of targets.
 - Update cost \sim target movement distance.
 - Query cost \sim **Area of R** .

Our Solution

- Use **differential forms** for tracking and answering **range queries** of mobile targets.
 - Decentralized.
 - Update cost \sim target movement distance
 - Query cost \sim **Perimeter of R**
 - Robust to node failures, link dynamics, mobility, coverage holes, sensing errors, location inaccuracies.

Using Range Queries for Target Tracking

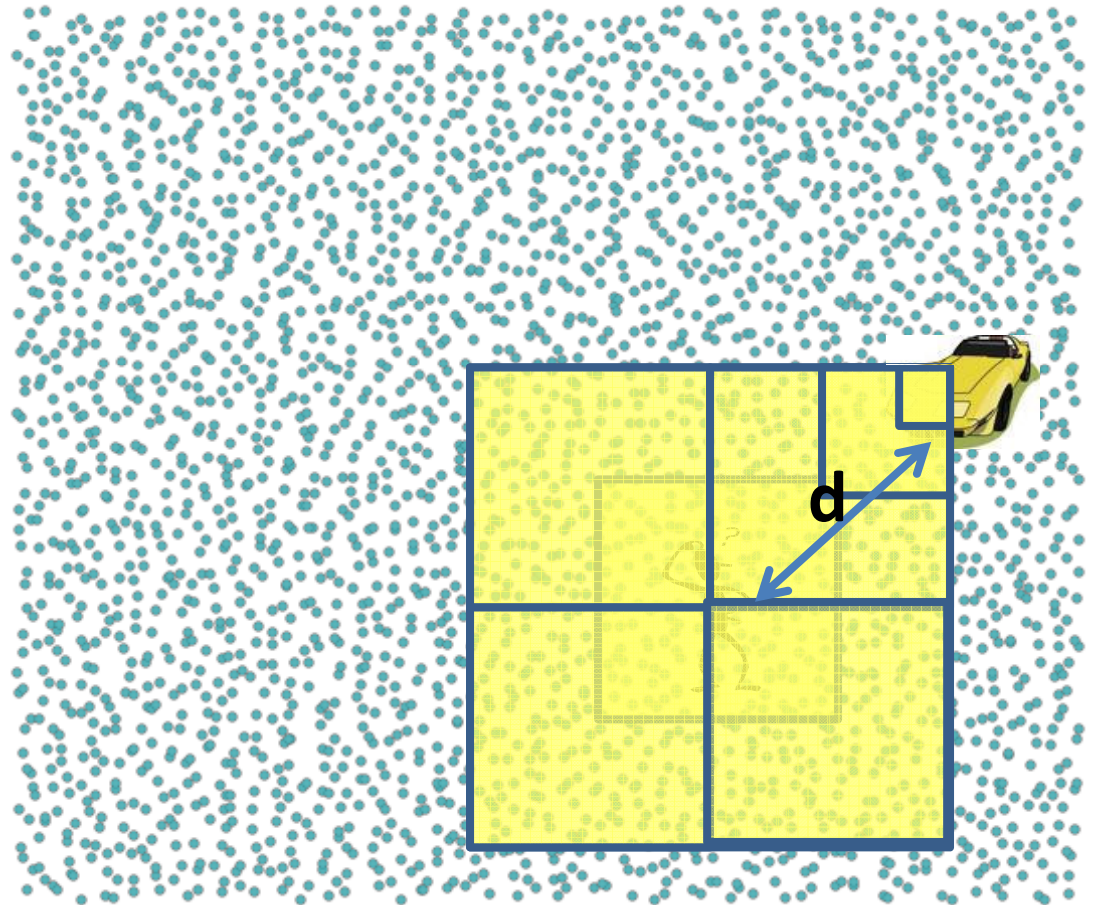
Find the yellow car?

1. Exponentially expand the range

– Stops when the range contains the target

2. Recurse & refine

- Total cost = $O(d)$
- Distance sensitive queries



$$1+2+4+8+\dots+d/2+d < 2d$$

Outline

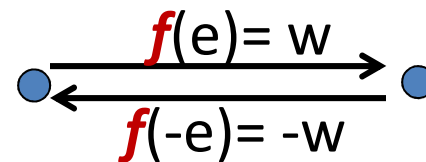
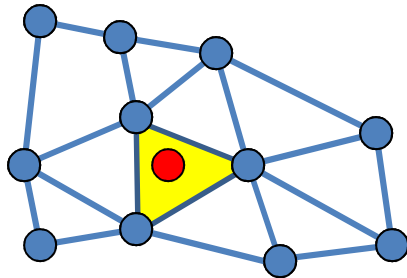
- Differential 1-form
- Algorithms for computing 1-form
- Network complications:
 - Dynamics: node/edge failures
 - Sensing errors
 - Network coverage holes
- Simulations and comparisons

Differential 1-Form

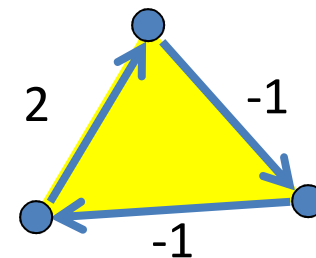
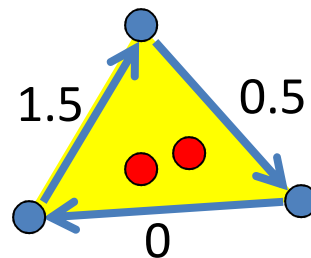
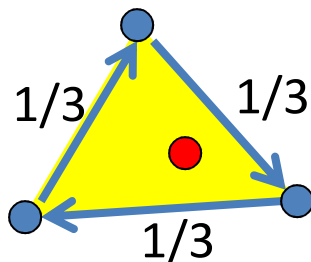
- A function defined on edges of a planar graph
 - Integrating the edge weights on region boundary gives the total weight of the targets inside.
1. Extract a planar graph.
 2. Range query: walk along region boundary and sum up edge weights.
 3. Target movement: change edge weight when a target crosses an edge

Definition: Differential 1-Form

- Planar graph G : a target stays within a **face**.
- Maintain “**directed**” weights f on edges.



- For each face, summing up weights **clockwise** gives the total weights of targets inside:



Boundary Operator

- Formally, a boundary operator applies on a face and returns the sum of the boundary edges in clockwise directions.

$$\partial \left[\begin{array}{c} \text{1/3} \quad \text{1/3} \\ \triangle \\ \text{1/3} \end{array} \right] = \begin{array}{c} \text{1/3} \quad \text{1/3} \\ \triangle \\ \text{1/3} \end{array}$$

- Extend f to a face.

$$f \left[\begin{array}{c} \text{1/3} \quad \text{1/3} \\ \triangle \\ \text{1/3} \end{array} \right] = f \left[\partial \begin{array}{c} \text{1/3} \quad \text{1/3} \\ \triangle \\ \text{1/3} \end{array} \right] = f \left[\begin{array}{c} \text{1/3} \quad \text{1/3} \\ \triangle \\ \text{1/3} \end{array} \right] = 1$$

Boundary Operator

- Boundary operator on a union of faces.

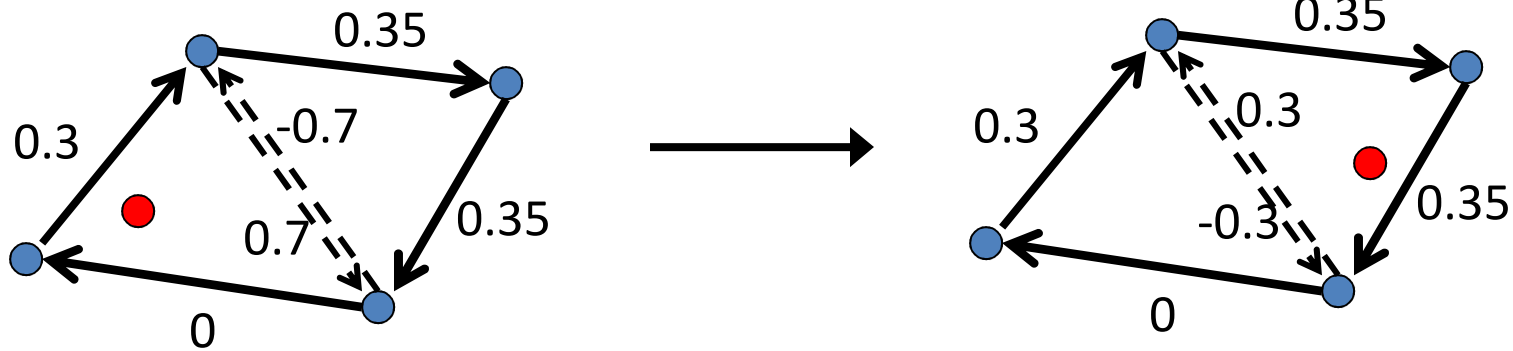
$$\begin{aligned} \partial [\text{yellow triangle} \cup \text{orange triangle}] &= \partial [\text{yellow triangle}] + \partial [\text{orange triangle}] \\ &= \begin{array}{c} \text{a} \nearrow \\ \text{b} \searrow \\ \text{c} \longleftarrow \end{array} + \begin{array}{c} \text{e} \longrightarrow \\ \text{d} \searrow \\ \text{-b} \longleftarrow \end{array} \\ &= \begin{array}{c} \text{a} \nearrow \\ \text{e} \longrightarrow \\ \text{c} \longleftarrow \\ \text{d} \searrow \end{array} \end{aligned}$$

Differential 1-Form for Range Queries

- Theorem: the total weights of targets inside a region \mathbf{R} is the sum of edge weights of $\partial\mathbf{R}$.
 - \mathbf{R} is simply a collection of faces, possibly disconnected.
- Range query: walk along $\partial\mathbf{R}$ in clockwise order and sum up edge weights.

Update 1-Form When Targets Move

- If a target crosses an edge e , subtract target weight from $f(e)$.



Communication Cost

- Assuming sensors have constant density.
- Update cost = # edges crossed
= $O(\text{distance moved})$
- Query cost = # edges on ∂R
= $O(\text{perimeter of } R)$

Multiple Targets

- Counting range query only
 - Maintain a single 1-form for all targets
- Queries for identifiable targets
 - Maintain a 1-form for each target.
- Maintain 1-form for each identifiable family of targets.
 - E.g., all cabs, all police cars, etc.
 - # cabs in the neighborhood, find a nearby cab

Outline

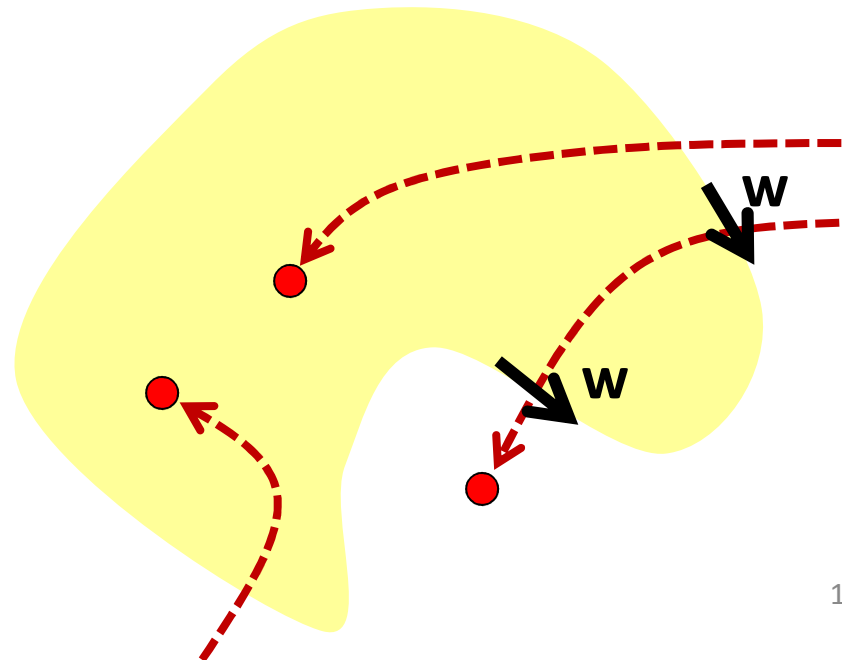
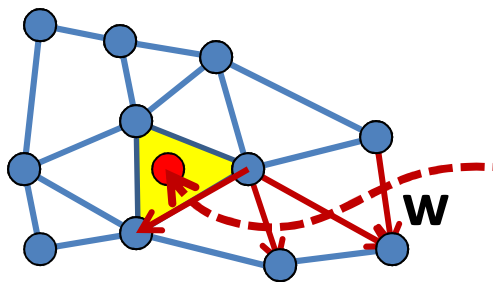
- Differential 1-form
- Algorithms for computing 1-form
- Network complications:
 - Dynamics: node/edge failures
 - Sensing inaccuracies
 - Network holes
- Simulations and comparisons

Extract a Planar Graph

- Extract a planar graph from connectivity graph
 - Location-based schemes
 - e.g., [Gao, et al, 01] [Sarkar et al, 09]
 - Location-free schemes
 - e.g., [Funke, Milosavljevic 07] [Zhang et. al, 08]
- Virtual planar graph
 - Only requirement: tell whether a target is within a face or not.

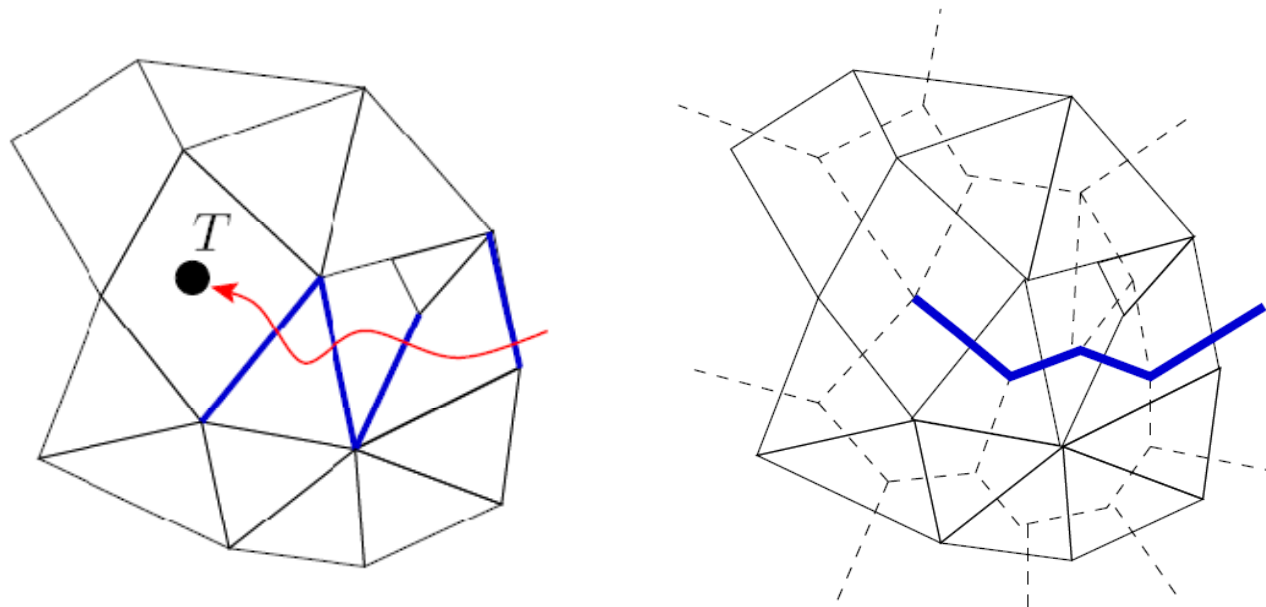
Initializing Differential 1-Form

- New targets coming into the network
 - Simply update f when a target comes in.
- For existing targets:
 - Imagine the target enters from the **face of infinity** along **any path**.



Multiple Targets: Sweep the Network

- Find a spanning tree T' of the dual graph G' , rooted at the face of infinity.
- Aggregate the weight of edges on T' .
- Weight of an edge in the primal = weight of the dual edge.
- Total communication cost for initialization = $O(n)$

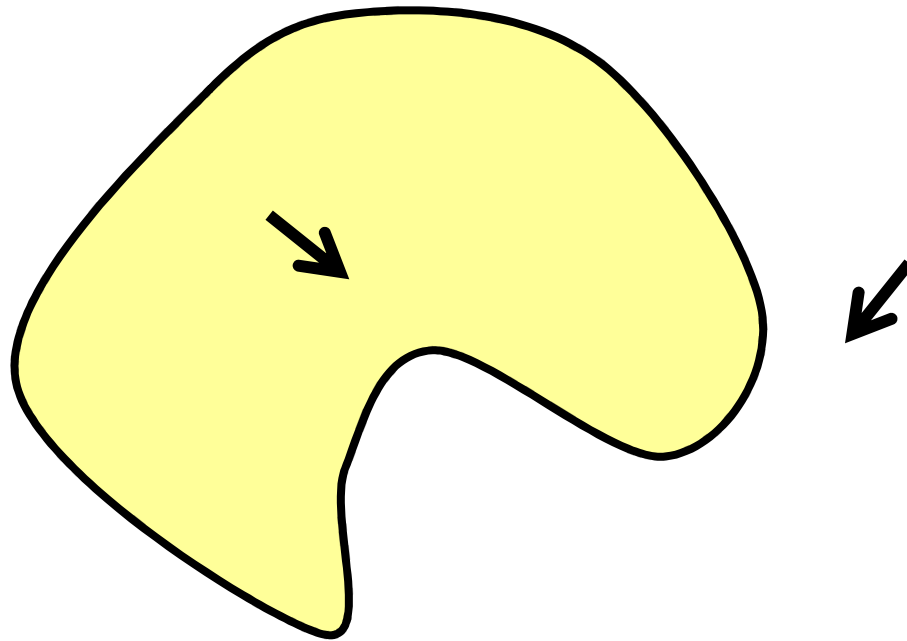


Outline

- Differential 1-form
- Algorithms for initializing 1-form
- **Network complications:**
 - Dynamics: node/edge failures
 - Sensing inaccuracies
 - Network holes
- Simulations and comparisons

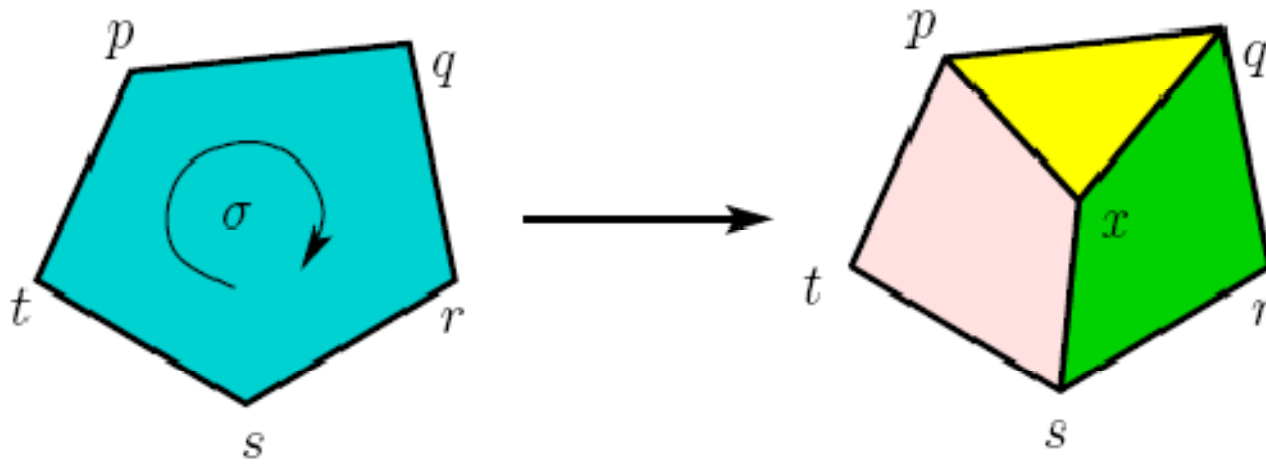
Robustness to Link or Node Failures

- A link failure or node failure in the interior or exterior of R does not affect the query result.



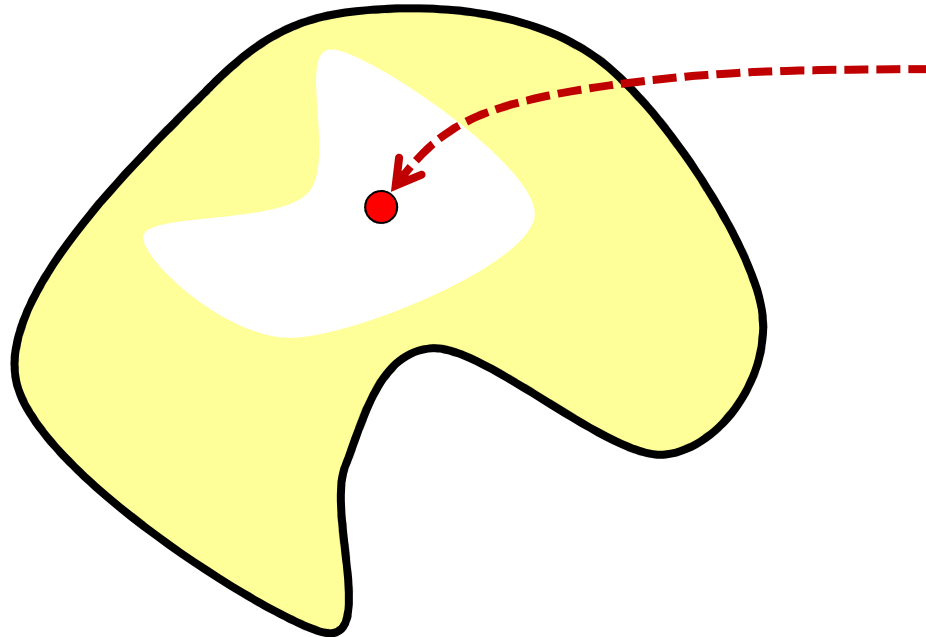
Robustness to Node Insertion

- Node insertion: refine the current face and give proper weights to the new edges.
- The weight of existing edges are not affected.



Robustness to Coverage Holes

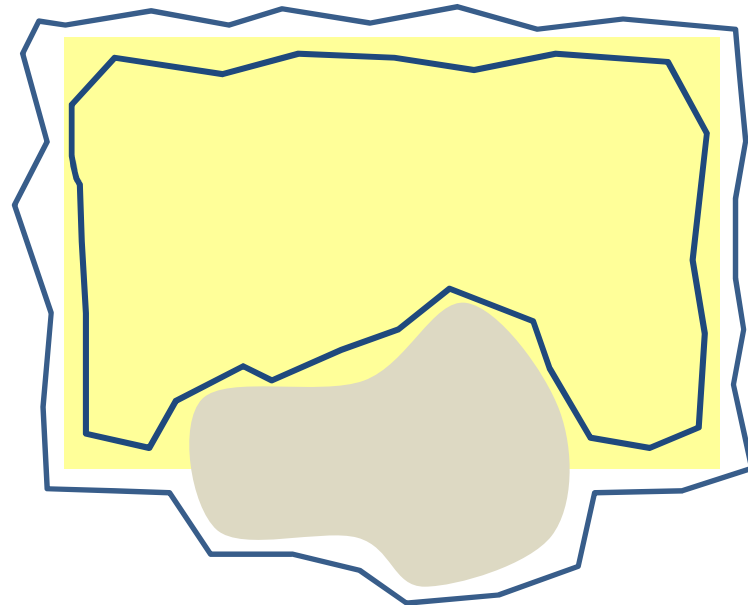
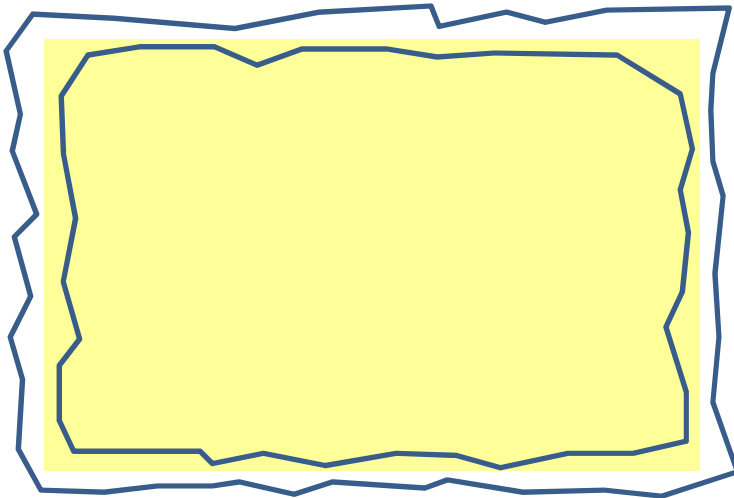
- A target can be lost in the hole but range query results of a region enclosing or disjoint of the hole are not affected.



Ranges Cutting Through Holes

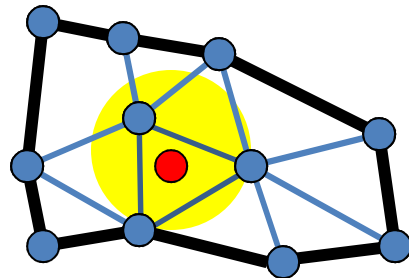
or, geometric ranges not following graph edges

- Take the best inner and outer approximation.
- Refine with detailed info from sensors near boundary



Robustness to Sensing Errors or Location Inaccuracies

- We are unsure of the precise target location but know the target is within a range.
- Any range query fully enclosing or disjoint with the target “feasible location” region gives correct results.



Tracking with Mobile Sensors

- Sensors can move.
 - Maintain the planar graph.
e.g., [Karp Kung 2001] [Gao, et al, 01]
 - When a target crosses an edge, update the 1-form.

Range Query of Continuous Data Fields

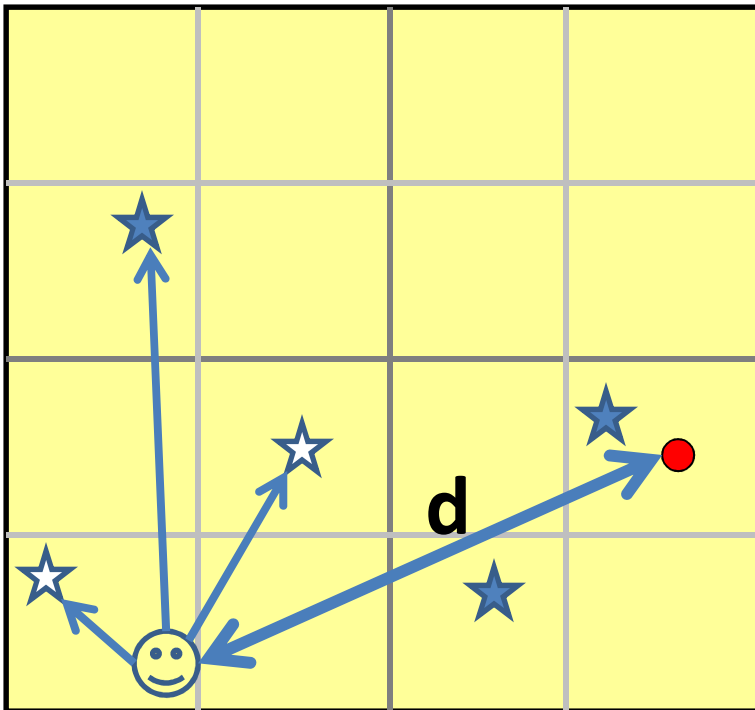
- Sensors monitor a temperature field.
- Treat each sensor reading as a target with certain weight.
- Apply the same scheme.

Outline

- Differential 1-form
- Algorithms for initializing 1-form
- Network complications:
 - Dynamics: node/edge failures
 - Sensing inaccuracies
 - Network holes
- **Simulations and comparisons**
 - Compare with location services
 - Robustness to link failures and sensing errors

Comparison with Location Services

- LLS [Abraham et al 2004]
 - Track a mobile target
 - Distributed hash table with hierarchical partitions



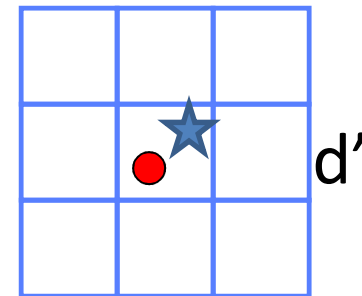
Tracked by one (hashed) location server at each square containing it at each level.

Query goes to the (hashed) location server at each square containing the query node.

Query cost : $O(d)$

Location Services

- LLS: lazy updates
 - A target does not trigger updates unless it moves outside the 9 squares.
 - The cost is $O(d')$.
 - The distance travelled is $\Omega(d')$.
 - Do this for each level



- Total update cost: $O(d' \log d')$ amortized, where d' is the movement distance.

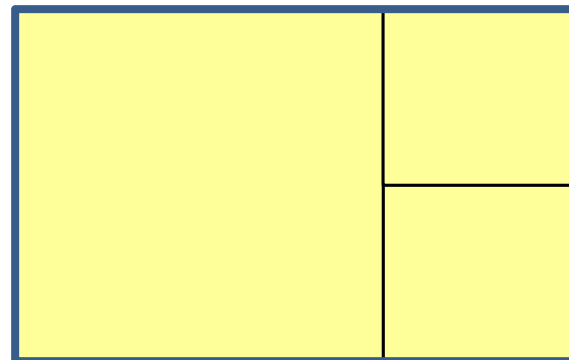
Differential Forms v.s. LLS

Differential form

- Designed for range queries
- Use recursive search for tracking query

LLS

- Designed for tracking query
- Use recursive search for range query --- query maximum quads within R.

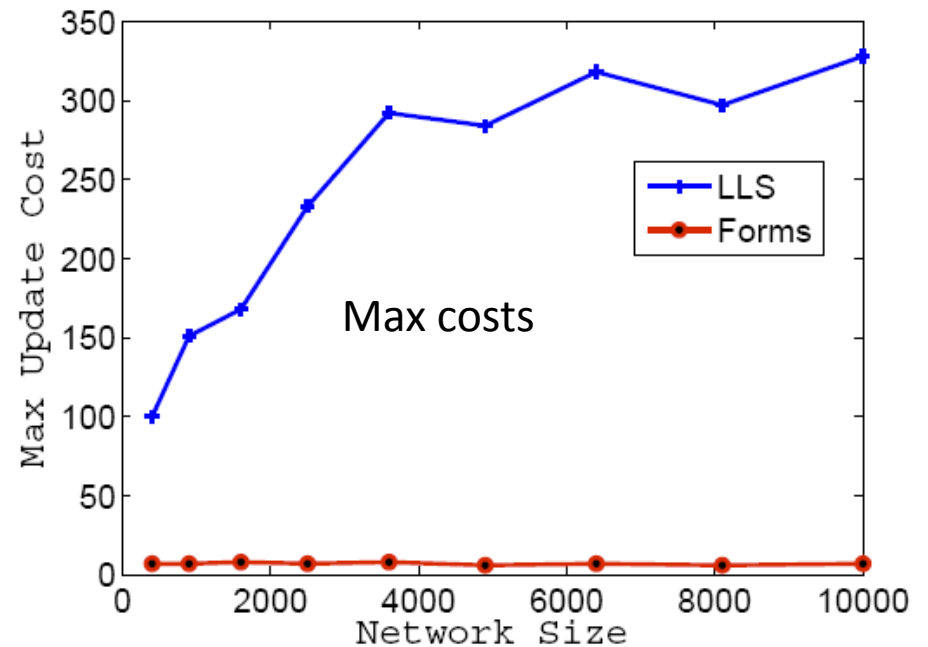
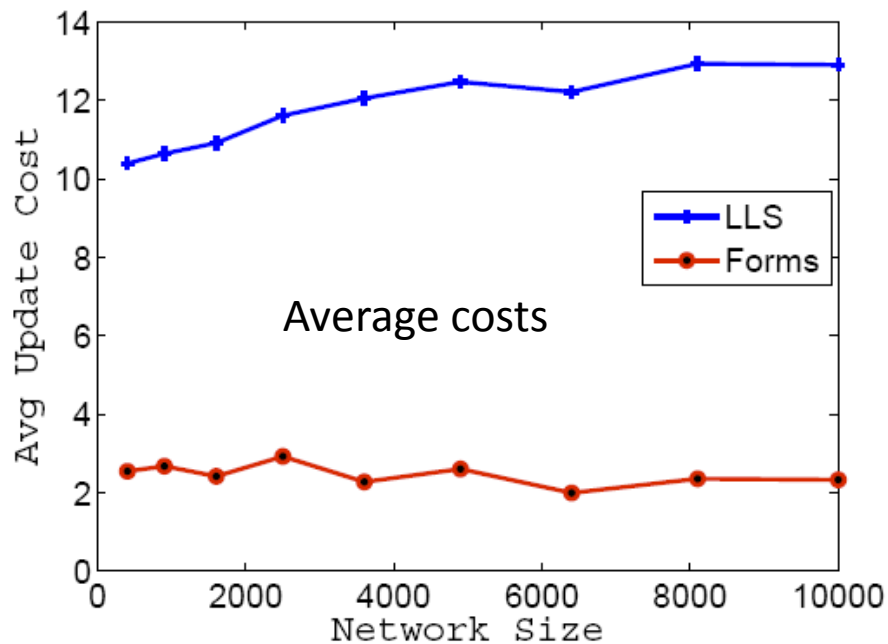


Differential Forms v.s. LLS

- Range query cost
Differential forms \ll LLS
- Query individual targets
Differential forms $\sim 2 \cdot$ LLS
- Update cost
Differential forms \ll LLS
 $O(d)$ v.s. $O(d \log d)$

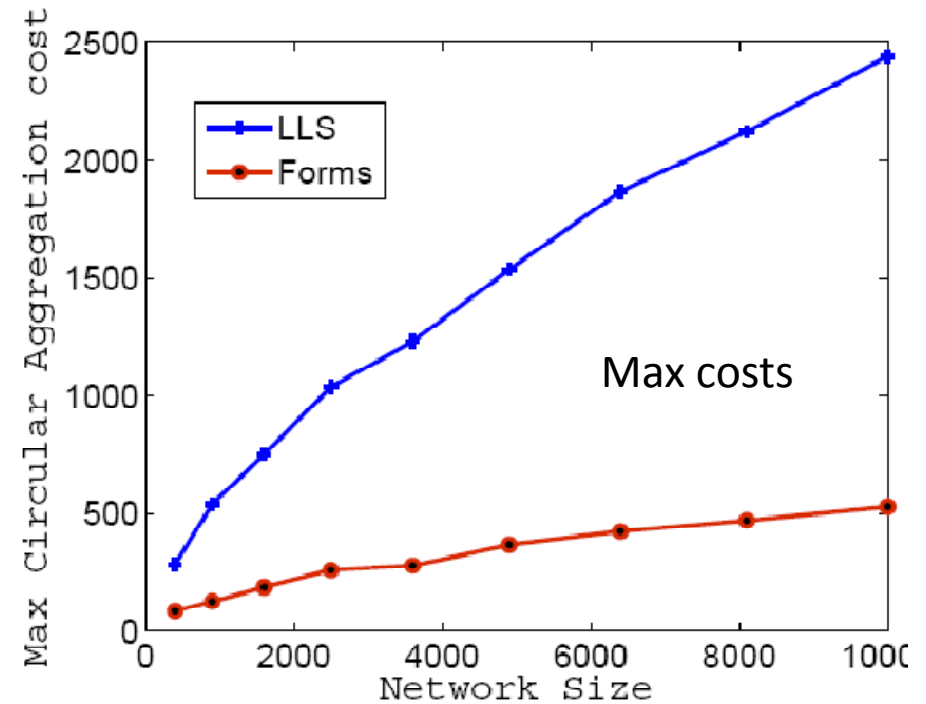
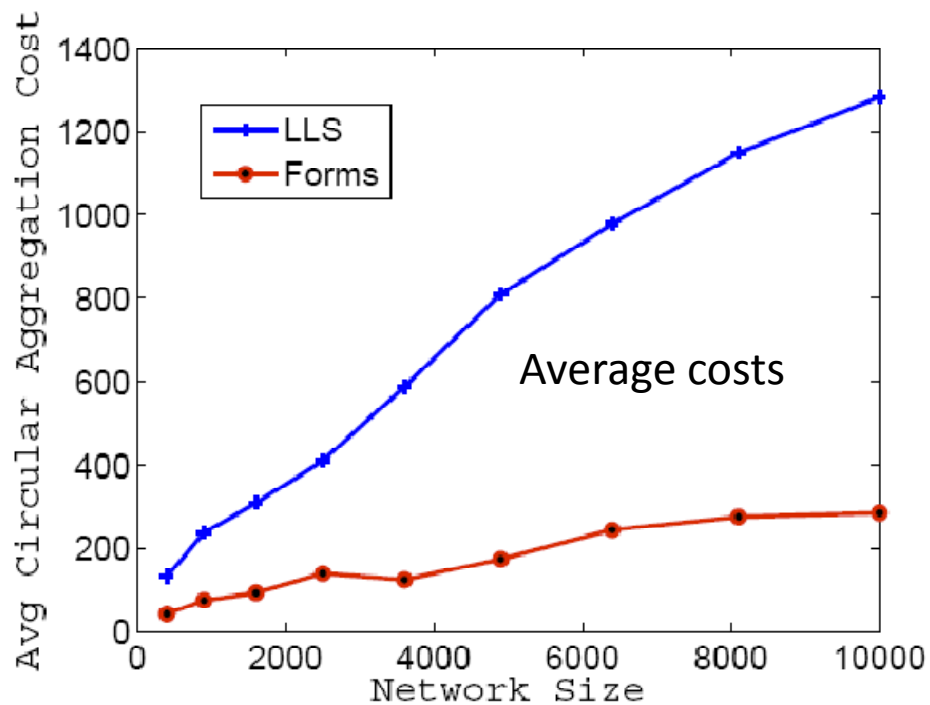
Update Costs

- The target moves one unit randomly per time unit --- discrete Brownian motion.
- LLS cost is amortized, some moves are expensive.



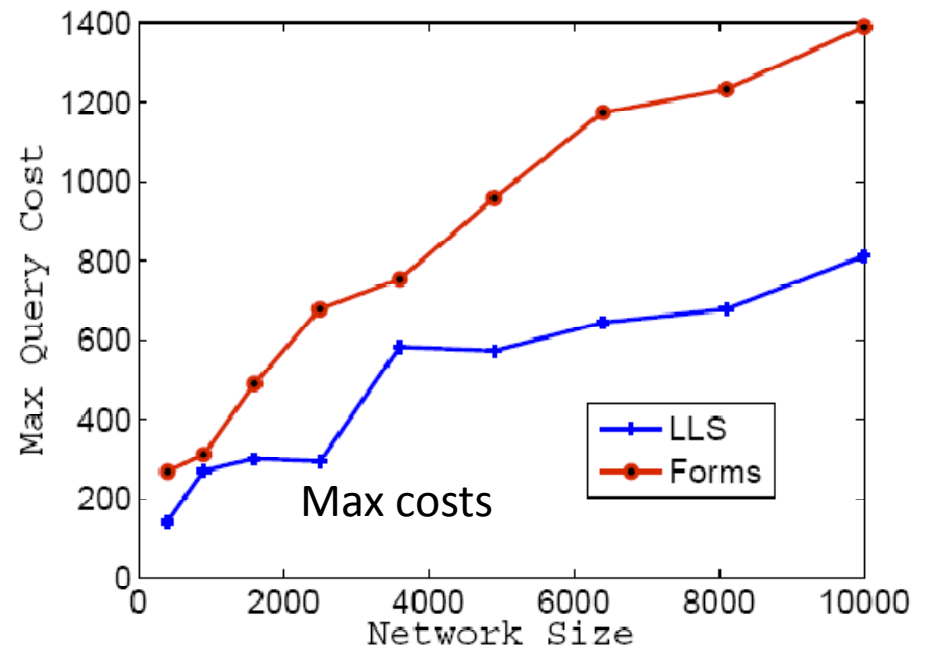
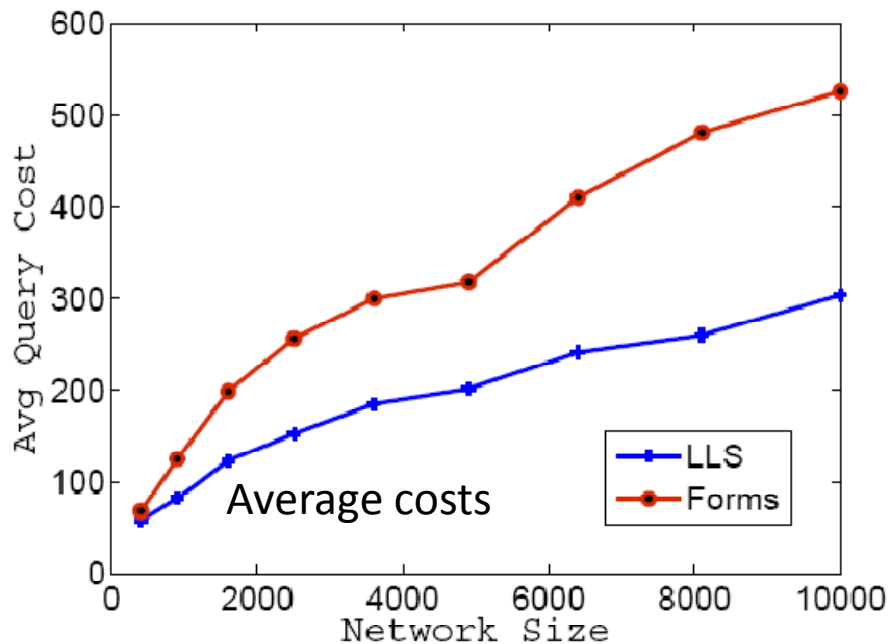
Range Queries Costs

- Ranges: random rectangles
- Caveats: for LLS we use the same hierarchy for all targets --- which saves query cost.



Tracking Query Costs

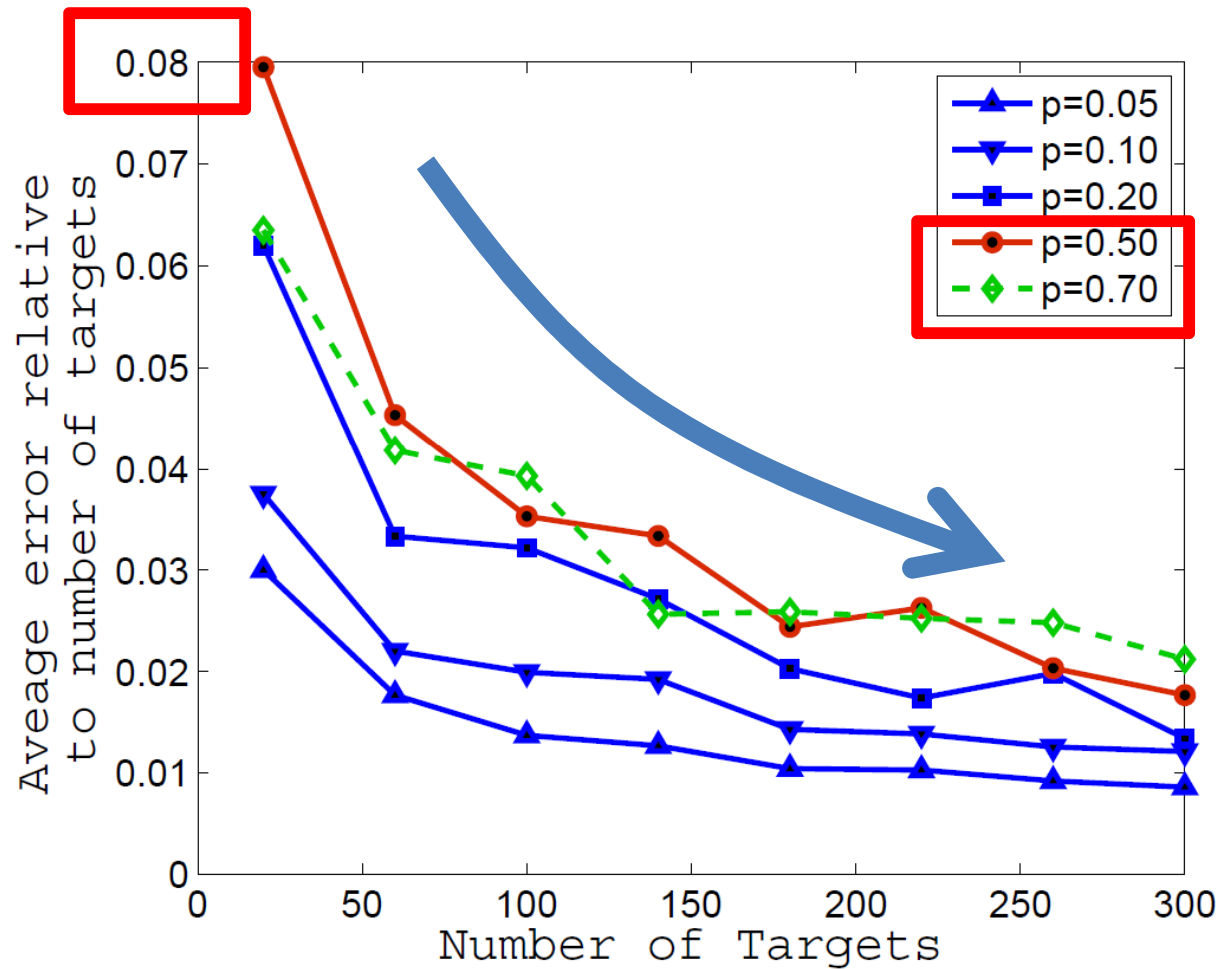
- Query for individual targets
- The expanding and refinement steps makes differential forms more costly.



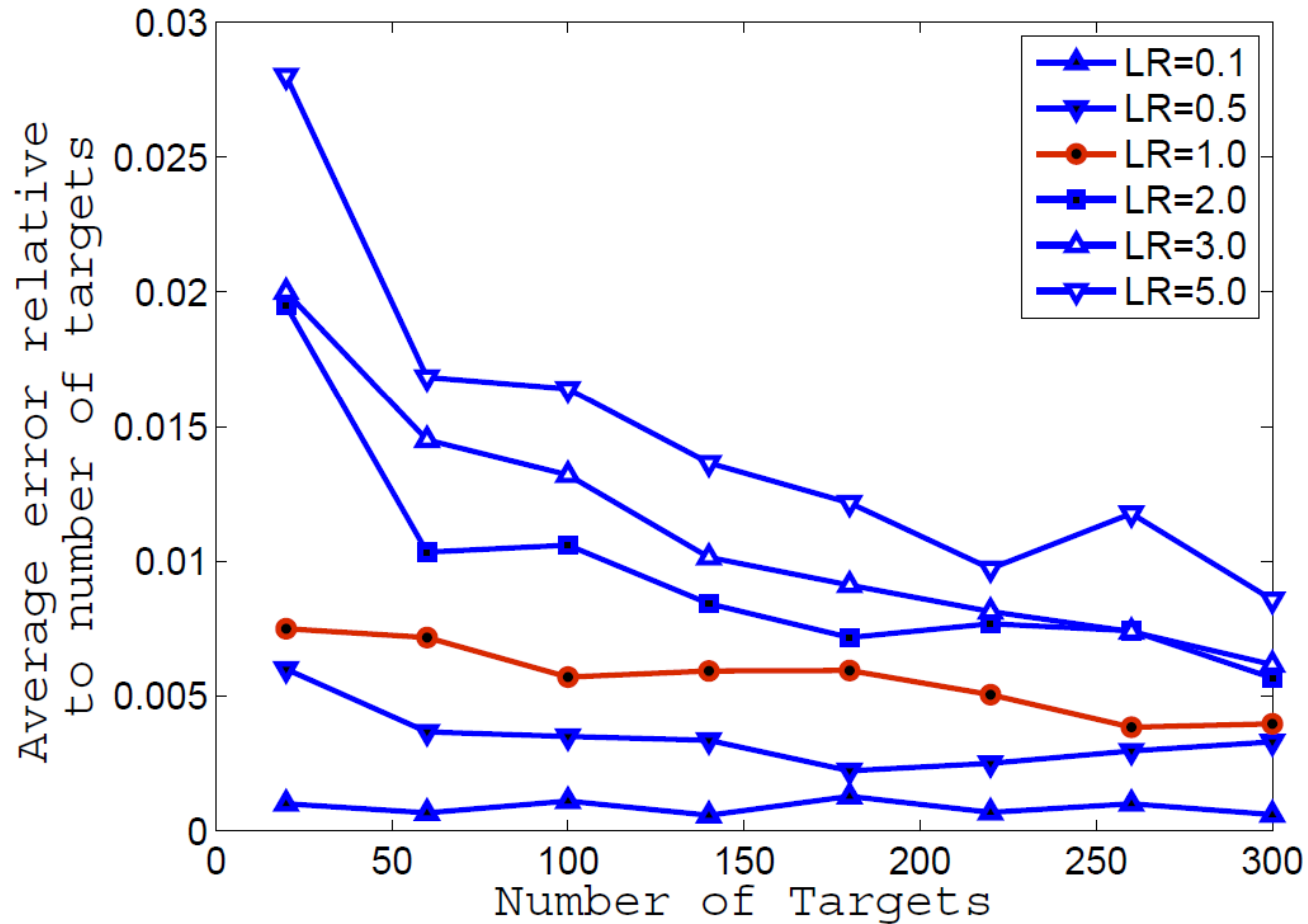
Target Detection Errors

- Fail to detect a target crossing an edge
 - Prob p : failure rate
- Target location error
 - LR: max distance of estimated loc from true loc
- Ranges: random axis-parallel rectangles
- Relative error = error in counts/ # targets

Robustness to Crossing Errors



Robustness to Sensor Location Errors



- Overcounting and undercounting cancel out.

Summary

- Differential form is a topological notion.
 - “Location-free” method
- Robust to network changes and sensing errors
- Sub-sampling sensors to conserve power by allowing gracefully degradation of query results.

Thank you!

- Questions and comments?